**Promise (ES6)**

технология для работы с асинхронными операциями

к примеру setTimeout, или при запросе на сервер

можно сделать через callback функции, но это все превращается в нечитаемый код (callback hell).

Links:

<https://learn.javascript.ru/promise-basics>

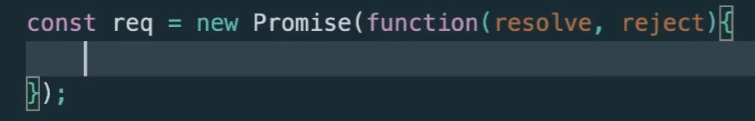
<https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Promise>

<https://stasonmars.ru/javascript/promisy-v-javascript-dlya-chainikov/>

new Promise(<callback function>)

методы **then(resolve функцию)** – если все ок**, catch(reject)**, **finally(obj)** – выполняется всегда

Promise.all(<arrayPromises>), Promise.rase(<arrayPromises>)



resolve и reject – функции, которые мы можем передавать (можно передавать только resolve)

**resolve** – означет что обещание выполнилось как мы ожидали

**reject** – значит наше обещание не выполнилось

Можно создавать цепочку (chaining) промисов (!преимущество от callback)



Можно вернуть, к примеру, data.



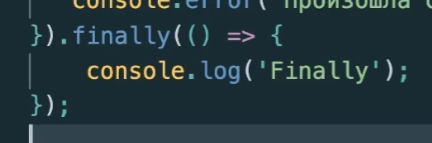
**Reject**

Когда к примеру ссылаемся на несуществующий сервер

Обычно **catch** в конце



**Finally**



**All**



После выполнения всех промисов

выполняется код из метода

**Race**

Отрабатывает когда у нас отработал первый Promise

